# Contributing to OctoberCMS

Thank you for your interest in contributing to the OctoberCMS project. We appreciate any assistance that community members and users of OctoberCMS are willing to provide. You can contribute to the project in several different ways:

- [Reporting a Security Vulnerability](#reporting-a-security-vulnerability)

- [Reporting an issue with OctoberCMS](#reporting-an-issue-with-octobercms)

- [Reporting an issue with an OctoberCMS plugin](#reporting-an-issue-with-an-octobercms-plugin)

- [Making a Feature Request](#making-a-feature-request)

- [Making a Pull Request](#making-a-pull-request)

- [Testing Pull Requests](#testing-a-pull-request)

## Reporting a Security Vulnerability

Please review [our security policy](https://github.com/octobercms/october/security/policy) on how to report security vulnerabilities.

## Reporting an issue with OctoberCMS

>\*\*NOTE:\*\* If your issue is related to an OctoberCMS plugin, please see the [Reporting an issue with an OctoberCMS plugin](#reporting-an-issue-with-an-octobercms-plugin) section below.

We work hard to process bugs that are reported, to assist with this please ensure the following details are always included:

- \*\*Summary\*\*: Make sure your summary reflects what the problem is and where it is. Provide as much detail as possible, the more information we have to work with the more likely it is that your problem can be solved.

- \*\*Reproduce steps\*\*: Clearly mention the steps to reproduce the bug.

- \*\*Expected behavior\*\*: Describe how OctoberCMS should behave on above mentioned steps.

- \*\*Actual behavior\*\*: What is the actual result on running above steps i.e. the bug behavior - \*\*include any error messages\*\*.

If possible, please provide any screenshots or GIFs of the issue occurring to provide us with additional context to determine the cause of the issue.

>\*\*NOTE\*\*: If you're reporting an issue that you intend to fix yourself, you can skip the Issue step and just submit a Pull Request that fixes the issue (along with a detailed description of the original problem) instead.

#### Here's how to report an issue on GitHub

1. \*\*Register for an account on [GitHub](https://github.com),\*\* if you don't already have one.

2. \*\*Search for similar issues.\*\* Perhaps someone has already reported your issue! If so, please add clarification and/or more information to the \*\*existing\*\* issue.

3. \*\*Create a new issue.\*\* If you don't find any similar issues, you can report your issue via the ["issues" tab](https://github.com/octobercms/october/issues) once you've logged in.

4. \*\*Monitor your issue and respond to questions.\*\* It may be necessary to provide additional information upon request or you might be asked if the issue still occurs after an update.

5. \*\*Close your issue.\*\* In case you notice that the issue doesn't occur anymore, you can close the issue yourself (don't forget to add a note saying that the issue is resolved and ideally any additional information on how it was resolved).

If you find out your bug is actually a duplicate of another bug and only notice that after you created it, please also close your bug with a short reference to the other issue that was there before.

#### Reporting an issue with an OctoberCMS plugin

>Please don't use the main GitHub for reporting issues with plugins.

If you have found a bug in a plugin, the best place to report it is with the [plugin author](https://octobercms.com/plugins).

If you are unable to contact the plugin author and the issue prevents the plugin from being used correctly, please feel free to email `hello@octobercms.com`, mentioning the plugin name, URL and the issue found. We will then determine if the plugin needs to be delisted.

#### Escalation process

We do our best to attend to all reported issues. If you have an important issue that requires attention, consider submitting a bounty using the [OctoberCMS Bounty Program](https://www.bountysource.com/teams/october).

## Making a Feature Request

>\*\*NOTE:\*\* Please don't use GitHub issues for suggesting a new feature. If you have a feature idea, the best place to suggest it is the [OctoberCMS website forum](https://octobercms.com/forum/chan/feature-requests).

Only use GitHub if you are planning on contributing a new feature and developing it. If you want to discuss your idea first, before "officially" posting it anywhere, you can always join us on [IRC](https://octobercms.com/chat) or [Slack](https://octobercms.slack.com).

#### GitHub feature requests

Feature Requests submitted as GitHub Issues specifically mean \*"I'd like to see this feature, I'm going to be working on some code to implement it."\* It is more like a Pre-Pull Request, in which a developer signifies that he or she wants to see a feature implemented that they think would be really great, and they're committed to coding it.

It's a great way to launch discussions on the developer side of things because both the core team and the community developer get a chance to talk about the technical side of the feature implementation. It's a great way to exchange ideas about how the logic could work in code.

## Making a Pull Request

Your contributions to the project are very welcome. If you would like to fix a bug or propose a new feature, you can submit a Pull Request.

To help us merge your Pull Request, please make sure you follow these points:

- Describe the problem clearly in the Pull Request description

- Please make your fix on the `develop` branch. This makes merging much easier.

- Do not edit compiled asset files such as `october.css`, `framework.css`, etc. directly. Instead, edit the LESS files inside the `less/` directory and then run `php artisan october:util compile assets` from the project root and commit the changed LESS files as well as the recompiled asset files.

- For any change that you make, \*\*please also add a test case(s)\*\* in the `tests/unit` directory. This helps us understand the issue and make sure that it will stay fixed forever.

Thank you for your contributions!

#### Best practices

It is ideal to keep your development branch or fork synchronised with the core OctoberCMS `develop` branch when submitting Pull Requests, as this minimises the possibility of merge conflicts.

To keep in sync with OctoberCMS, add the core OctoberCMS repository as a Git remote (ie. `upstream`) and pull changes from the OctoberCMS repository into your local `develop` branch as often as possible:

```

git remote add upstream git@github.com:octobercms/october.git

git fetch upstream

git checkout develop

git pull upstream develop

```

This ensures that your local `develop` branch matches OctoberCMS. When developing a pull request, it is best to use your own development branch. For example, creating a fix to improve spelling on a language file could be made into a branch called `lang-en-spelling-fixes`, which can be branched off from the `develop` branch.

```

git checkout -b lang-en-spelling-fixes develop

```

When you wish to update your development branch with the latest changes from the `develop` branch, it is just a simple merge:

```

git merge develop

```

This will merge all the latest changes from the OctoberCMS `develop` branch into your development branch.

#### Resolving merge conflicts

Occassionally, you may encounter a merge conflict with your Pull Request. This most commonly occurs if another change made to the OctoberCMS repository was made to a file that your Pull Request has also changed.

It is the responsibility of the author of the Pull Request to resolve any merge conflicts before their Pull Request is accepted.

You should ensure that your local copy of OctoberCMS is synchronised with with the `develop` branch in the OctoberCMS repository. Please follow the [steps above](#best-practices) to synchronise the repositories.

If Git reports that your changes have conflicts, you will need to resolve the changes in a way that includes the changes from the OctoberCMS repository as well as implementing your Pull Request's changes. See GitHub's guide to [resolving a merge conflict](https://help.github.com/en/articles/resolving-a-merge-conflict-using-the-command-line) for tips on resolving conflicts.

#### PSR Coding standards

Please ensure that your Pull Request satisfies the following coding standards:

- [PSR 2 Coding Style Guide](https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-2-coding-style-guide.md)

- [PSR 1 Coding Style Guide](https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-1-basic-coding-standard.md)

- [PSR 0 Coding Style Guide](https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-0.md)

To validate your changes against our coding standards, you may run `./vendor/bin/phpcs -nq --extensions="php"` in your development folder.

#### Team rules

The OctoberCMS team follows the [developer guidelines](https://octobercms.com/docs/help/developer-guide) as much as possible.

## Testing a Pull Request

Although we aim to test all pull requests made to the OctoberCMS repository, the maintainers of OctoberCMS are volunteers and may not be able to promptly attend to all pull requests.

To help speed things up, any assistance with testing Pull Requests and fixes will be very appreciated. The best Pull Requests to test are those that are tagged as [\*\*Testing Needed\*\*](https://github.com/octobercms/october/pulls?q=is%3Apr+is%3Aopen+label%3A%22Testing+Needed%22) in the repository.

To test a Pull Request, you can use the steps below in a terminal or command-line interface to create a fresh installation of OctoberCMS with the changes made in the Pull Request, ready to test. In this example, we have a user called `qwerty123` that has created a pull request with an ID of `#4509`.

1. Check out a copy of the OctoberCMS repository to a folder that you can view in your web browser: `git clone git@github.com:octobercms/october.git`. This will add the files into a subfolder called `october`.

2. Then, go to the `october` subfolder and check out \*\*@qwerty123\*\*'s changes in a branch in your local repository: `git fetch origin pull/4509/head:pr-4509`. This will pull their changes into a branch called `pr-4509`. You will then need to check out the branch: `git checkout pr-4509`.

3. Next, get the Composer dependencies: `composer update`.

4. Next, run `php artisan october:env` to create a `.env` file in your folder. This will contain the configuration values for the database and site.

5. Finally, once you've populated that file with your database and site details, run `php artisan october:up` to install the necessary database tables.

At this point, you should have a working copy of the Pull Request ready to test.